# Aero-Fit Business Case

In [1]:
```
# Importing the libraries

import pandas as pd
import numpy as np
```

In [2]:
```
# Loading the dataset

data = pd.read_csv("aerofit_treadmill.csv")
```

In [3]:
```
# Checking the Data

data.head(10)
```

Out[3]:

| | Product | Age | Gender | Education | MaritalStatus | Usage | Fitness | Income | Miles |
|---|---|---|---|---|---|---|---|---|---|
| **0** | KP281 | 18 | Male | 14 | Single | 3 | 4 | 29562 | 112 |
| **1** | KP281 | 19 | Male | 15 | Single | 2 | 3 | 31836 | 75 |
| **2** | KP281 | 19 | Female | 14 | Partnered | 4 | 3 | 30699 | 66 |
| **3** | KP281 | 19 | Male | 12 | Single | 3 | 3 | 32973 | 85 |
| **4** | KP281 | 20 | Male | 13 | Partnered | 4 | 2 | 35247 | 47 |
| **5** | KP281 | 20 | Female | 14 | Partnered | 3 | 3 | 32973 | 66 |
| **6** | KP281 | 21 | Female | 14 | Partnered | 3 | 3 | 35247 | 75 |
| **7** | KP281 | 21 | Male | 13 | Single | 3 | 3 | 32973 | 85 |
| **8** | KP281 | 21 | Male | 15 | Single | 5 | 4 | 35247 | 141 |
| **9** | KP281 | 21 | Female | 15 | Partnered | 2 | 3 | 37521 | 85 |

In [4]:
```
# Shape of the dataset

data.shape
```

Out[4]:
```
(180, 9)
```

In [5]:
```
# Checking for the basic info

data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180 entries, 0 to 179
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   Product        180 non-null    object
 1   Age            180 non-null    int64 
 2   Gender         180 non-null    object
 3   Education      180 non-null    int64 
 4   MaritalStatus  180 non-null    object
 5   Usage          180 non-null    int64 
 6   Fitness        180 non-null    int64 
 7   Income         180 non-null    int64 
 8   Miles          180 non-null    int64 
dtypes: int64(6), object(3)
memory usage: 12.8+ KB
```

In [6]:
```python
# Checking for Datatypes

data.dtypes
```

Out[6]:

| | 0 |
|---|---|
| **Product** | object |
| **Age** | int64 |
| **Gender** | object |
| **Education** | int64 |
| **MaritalStatus** | object |
| **Usage** | int64 |
| **Fitness** | int64 |
| **Income** | int64 |
| **Miles** | int64 |

**dtype:** object

In [7]:
```python
# Checking for NULL Values

data.isna().sum()
```

Out[7]:

| | 0 |
|---|---|
| **Product** | 0 |
| **Age** | 0 |
| **Gender** | 0 |
| **Education** | 0 |
| **MaritalStatus** | 0 |
| **Usage** | 0 |
| **Fitness** | 0 |
| **Income** | 0 |
| **Miles** | 0 |

**dtype:** int64

In [8]:
```
# Checking for Unique value counts

data.nunique()
```

Out[8]:

| | 0 |
|---|---|
| **Product** | 3 |
| **Age** | 32 |
| **Gender** | 2 |
| **Education** | 8 |
| **MaritalStatus** | 2 |
| **Usage** | 6 |
| **Fitness** | 5 |
| **Income** | 62 |
| **Miles** | 37 |

**dtype:** int64

### Checking for Uniques values for each column

In [9]:
```
data["Product"].unique()
```

Out[9]:
```
array(['KP281', 'KP481', 'KP781'], dtype=object)
```

In [10]:
```
data["Age"].unique()
```

Out[10]:
```
array([18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
       35, 36, 37, 38, 39, 40, 41, 43, 44, 46, 47, 50, 45, 48, 42])
```

In [11]:
```
data["Education"].unique()
```

Out[11]:
```
array([14, 15, 12, 13, 16, 18, 20, 21])
```

In [12]:
```
data["MaritalStatus"].unique()
```

Out[12]:
```
array(['Single', 'Partnered'], dtype=object)
```

In [13]:
```
data["Usage"].unique()
```

Out[13]:
```
array([3, 2, 4, 5, 6, 7])
```

In [14]:
```
data["Fitness"].unique()
```

Out[14]:
```
array([4, 3, 2, 1, 5])
```

In [15]:
```
data["Income"].unique()
```

Out[15]:
```
array([ 29562,  31836,  30699,  32973,  35247,  37521,  36384,  38658,
        40932,  34110,  39795,  42069,  44343,  45480,  46617,  48891,
        53439,  43206,  52302,  51165,  50028,  54576,  68220,  55713,
        60261,  67083,  56850,  59124,  61398,  57987,  64809,  47754,
        65220,  62535,  48658,  54781,  48556,  58516,  53536,  61006,
        57271,  52291,  49801,  62251,  64741,  70966,  75946,  74701,
        69721,  83416,  88396,  90886,  92131,  77191,  52290,  85906,
       103336,  99601,  89641,  95866, 104581,  95508])
```

In [16]:
```
data["Miles"].unique()
```

Out[16]:
```
array([112,  75,  66,  85,  47, 141, 103,  94, 113,  38, 188,  56, 132,
       169,  64,  53, 106,  95, 212,  42, 127,  74, 170,  21, 120, 200,
       140, 100,  80, 160, 180, 240, 150, 300, 280, 260, 360])
```

In [17]:
```
data["Product"].value_counts()
```

Out[17]:

| Product | count |
|---|---|
| KP281 | 80 |
| KP481 | 60 |
| KP781 | 40 |

**dtype:** int64

In [18]:
```
data["Age"].value_counts()
```

Out[18]:

| Age | count |
|---|---|
| 25 | 25 |
| 23 | 18 |
| 24 | 12 |
| 26 | 12 |
| 28 | 9 |
| 35 | 8 |
| 33 | 8 |
| 30 | 7 |
| 38 | 7 |
| 21 | 7 |
| 22 | 7 |
| 27 | 7 |
| 31 | 6 |
| 34 | 6 |
| 29 | 6 |
| 20 | 5 |
| 40 | 5 |
| 32 | 4 |
| 19 | 4 |
| 48 | 2 |
| 37 | 2 |
| 45 | 2 |
| 47 | 2 |
| 46 | 1 |
| 50 | 1 |
| 18 | 1 |
| 44 | 1 |
| 43 | 1 |
| 41 | 1 |
| 39 | 1 |
| 36 | 1 |
| 42 | 1 |

**dtype:** int64

In [19]:
```
data["Gender"].value_counts()
```

Out[19]:

| Gender | count |
|---|---|
| Male | 104 |
| Female | 76 |

**dtype:** int64

In [20]:
```
data["Education"].value_counts()
```

Out[20]:

| Education | count |
|---|---|
| 16 | 85 |
| 14 | 55 |
| 18 | 23 |
| 15 | 5 |
| 13 | 5 |
| 12 | 3 |
| 21 | 3 |
| 20 | 1 |

**dtype:** int64

In [21]:
```
data["MaritalStatus"].value_counts()
```

Out[21]:

| MaritalStatus | count |
|---|---|
| Partnered | 107 |
| Single | 73 |

**dtype:** int64

In [22]:
```
data["Usage"].value_counts()
```

Out[22]:

| Usage | count |
|---|---|
| 3 | 69 |
| 4 | 52 |
| 2 | 33 |
| 5 | 17 |
| 6 | 7 |
| 7 | 2 |

**dtype:** int64

In [23]:
```
data["Fitness"].value_counts()
```

Out[23]:

| Fitness | count |
|---|---|
| 3 | 97 |
| 5 | 31 |
| 2 | 26 |
| 4 | 24 |
| 1 | 2 |

**dtype:** int64

In [24]:
```
data.describe()
```

Out[24]:

| | Age | Education | Usage | Fitness | Income | Miles |
|---|---|---|---|---|---|---|
| count | 180.000000 | 180.000000 | 180.000000 | 180.000000 | 180.000000 | 180.000000 |
| mean | 28.788889 | 15.572222 | 3.455556 | 3.311111 | 53719.577778 | 103.194444 |
| std | 6.943498 | 1.617055 | 1.084797 | 0.958869 | 16506.684226 | 51.863605 |
| min | 18.000000 | 12.000000 | 2.000000 | 1.000000 | 29562.000000 | 21.000000 |
| 25% | 24.000000 | 14.000000 | 3.000000 | 3.000000 | 44058.750000 | 66.000000 |
| 50% | 26.000000 | 16.000000 | 3.000000 | 3.000000 | 50596.500000 | 94.000000 |
| 75% | 33.000000 | 16.000000 | 4.000000 | 4.000000 | 58668.000000 | 114.750000 |
| max | 50.000000 | 21.000000 | 7.000000 | 5.000000 | 104581.000000 | 360.000000 |

In [25]:
```
# Importing the required Libraries

import matplotlib.pyplot as plt
import seaborn as sns
```

In [26]:
```
#Replacing the String values to Numeric to create a correlation.

df = data.copy()
```

```
df["Product"].replace({"KP281" : 0 , "KP481" : 1 , "KP781" : 2}, inplace = True)
df["Gender"].replace({"Male" : 1 , "Female" : 0}, inplace = True)
df["MaritalStatus"].replace({"Single" : 0 , "Partnered" : 1}, inplace = True)

df.head()
```

Out[26]:

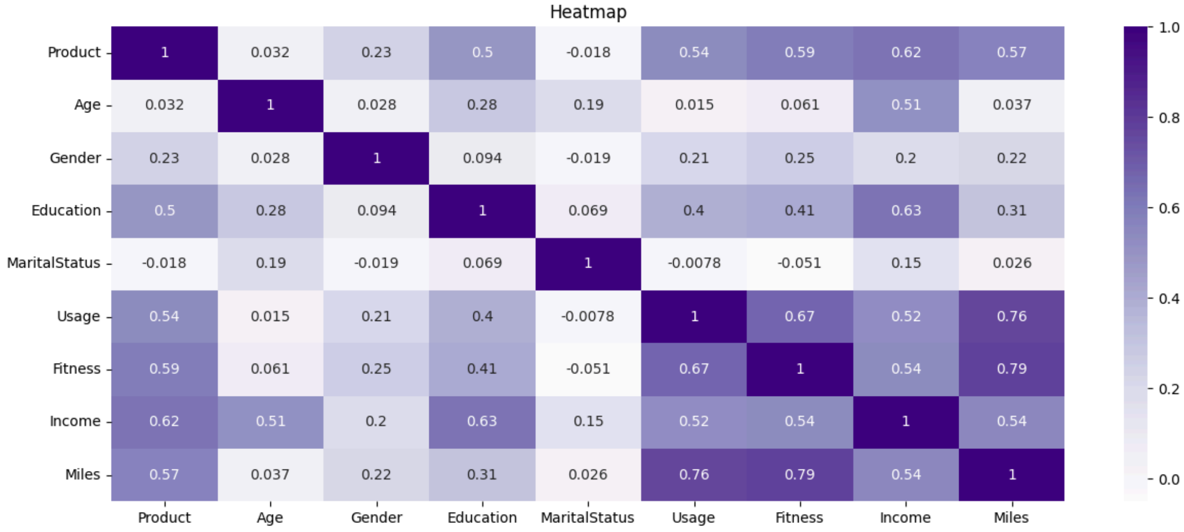
| | Product | Age | Gender | Education | MaritalStatus | Usage | Fitness | Income | Miles |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 18 | 1 | 14 | 0 | 3 | 4 | 29562 | 112 |
| **1** | 0 | 19 | 1 | 15 | 0 | 2 | 3 | 31836 | 75 |
| **2** | 0 | 19 | 0 | 14 | 1 | 4 | 3 | 30699 | 66 |
| **3** | 0 | 19 | 1 | 12 | 0 | 3 | 3 | 32973 | 85 |
| **4** | 0 | 20 | 1 | 13 | 1 | 4 | 2 | 35247 | 47 |

In [27]:
```
df.corr()
```

Out[27]:

| | Product | Age | Gender | Education | MaritalStatus | Usage | Fitness | Inco |
|---|---|---|---|---|---|---|---|---|
| **Product** | 1.000000 | 0.032225 | 0.230653 | 0.495018 | -0.017602 | 0.537447 | 0.594883 | 0.624 |
| **Age** | 0.032225 | 1.000000 | 0.027544 | 0.280496 | 0.192152 | 0.015064 | 0.061105 | 0.513 |
| **Gender** | 0.230653 | 0.027544 | 1.000000 | 0.094089 | -0.018836 | 0.214424 | 0.254609 | 0.202 |
| **Education** | 0.495018 | 0.280496 | 0.094089 | 1.000000 | 0.068569 | 0.395155 | 0.410581 | 0.625 |
| **MaritalStatus** | -0.017602 | 0.192152 | -0.018836 | 0.068569 | 1.000000 | -0.007786 | -0.050751 | 0.150 |
| **Usage** | 0.537447 | 0.015064 | 0.214424 | 0.395155 | -0.007786 | 1.000000 | 0.668606 | 0.519 |
| **Fitness** | 0.594883 | 0.061105 | 0.254609 | 0.410581 | -0.050751 | 0.668606 | 1.000000 | 0.535 |
| **Income** | 0.624168 | 0.513414 | 0.202053 | 0.625827 | 0.150293 | 0.519537 | 0.535005 | 1.000 |
| **Miles** | 0.571596 | 0.036618 | 0.217869 | 0.307284 | 0.025639 | 0.759130 | 0.785702 | 0.543 |

In [28]:
```
plt.figure(figsize=(15,6))
sns.heatmap(df.corr(), cmap = "Purples", annot=True)

plt.title("Heatmap")
plt.show()
```




**Insight :**

- The product bought has a higher correlation with income, fitness, miles walked and usage.
- Age has a higher correlation with income which obvious that as the age increases income can increase too.
- Gender has comparitively low correlation with Fitness, Product, Miles and Usage.
- Education is highly correlated with Income which is as expected followed by the product bought.
- Marital Status doesnt exhibit much correlation with any of the factors considered.
- Usage is very much highly correlated with Miles walked, usage, Product bought and income.
- Fitness has greater relation with Miles, Usage, Product and Income.

In [ ]:

```
# Analysing the Percentage of product purchased

code = data.groupby(data["Product"])["Product"].count()
len = data.shape[0]
pct = (code / len)*100

df = pd.DataFrame(pct)
df
```

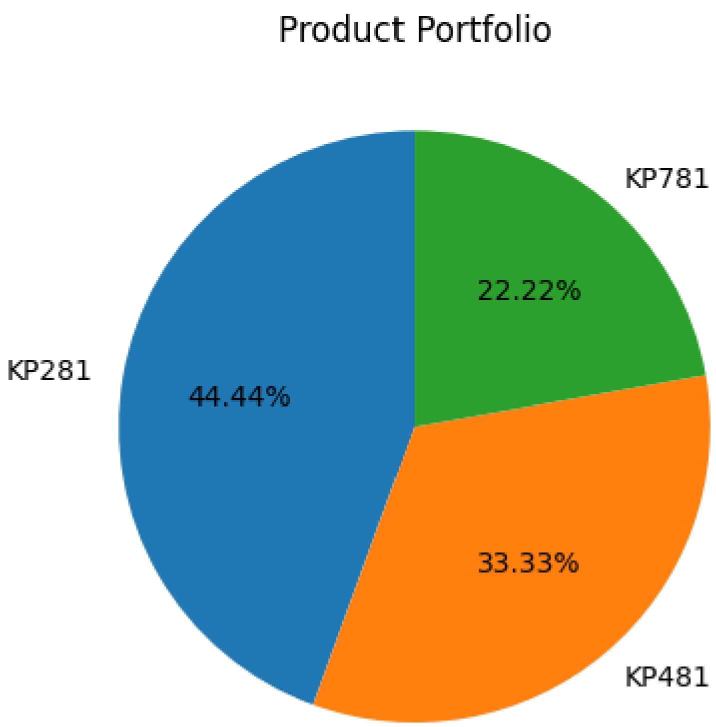
Out[ ]:

| | Product |
|---|---|
| **Product** | |
| **KP281** | 44.444444 |
| **KP481** | 33.333333 |
| **KP781** | 22.222222 |

In [ ]:

```
plt.pie(df["Product"],
        labels = df.index,
        startangle = 90,
```

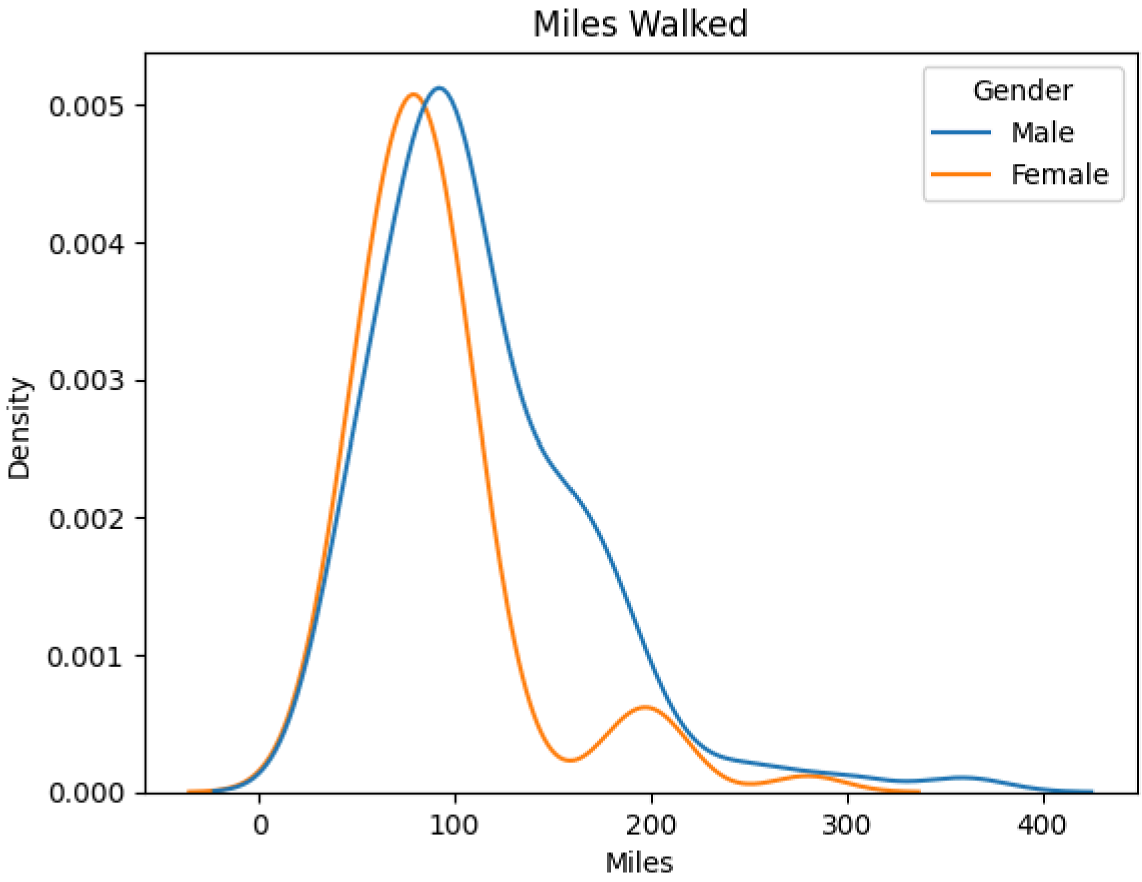
```
        autopct = "%.2f%%")

plt.title("Product Portfolio")
plt.show()
```




**Insight :**

- From the above pie chart its clear that 22.22% people bought KP781 , 33.33% bought KP481 , 44.44% bought KP281.
- Hence KP281 is the most purchased model.

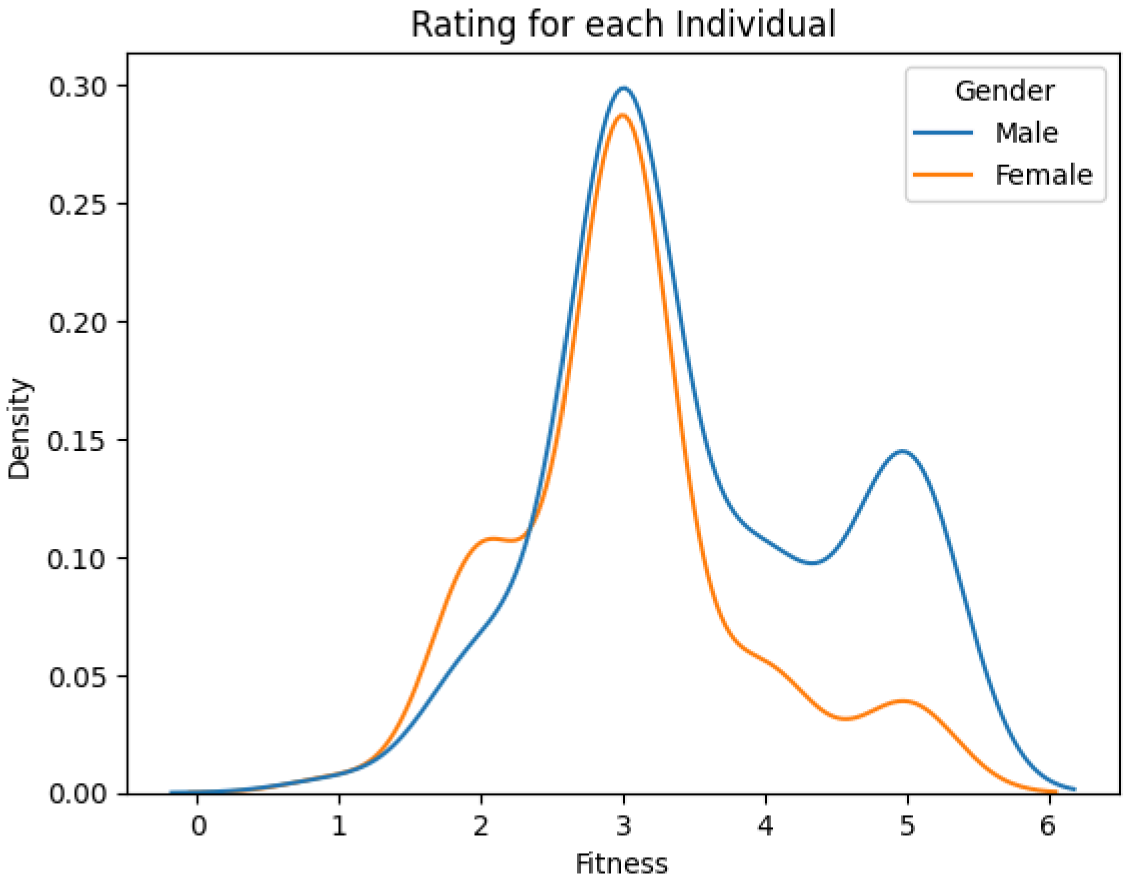
In [ ]:
```
# Analysing the Miles coved by the individuals

sns.kdeplot(data=data , x="Miles", hue="Gender")

plt.title("Miles Walked")
plt.show()
```




**Insight :**

- From the line plot its clear that most of Male and Female has walking mile betweeen 85 to 100.

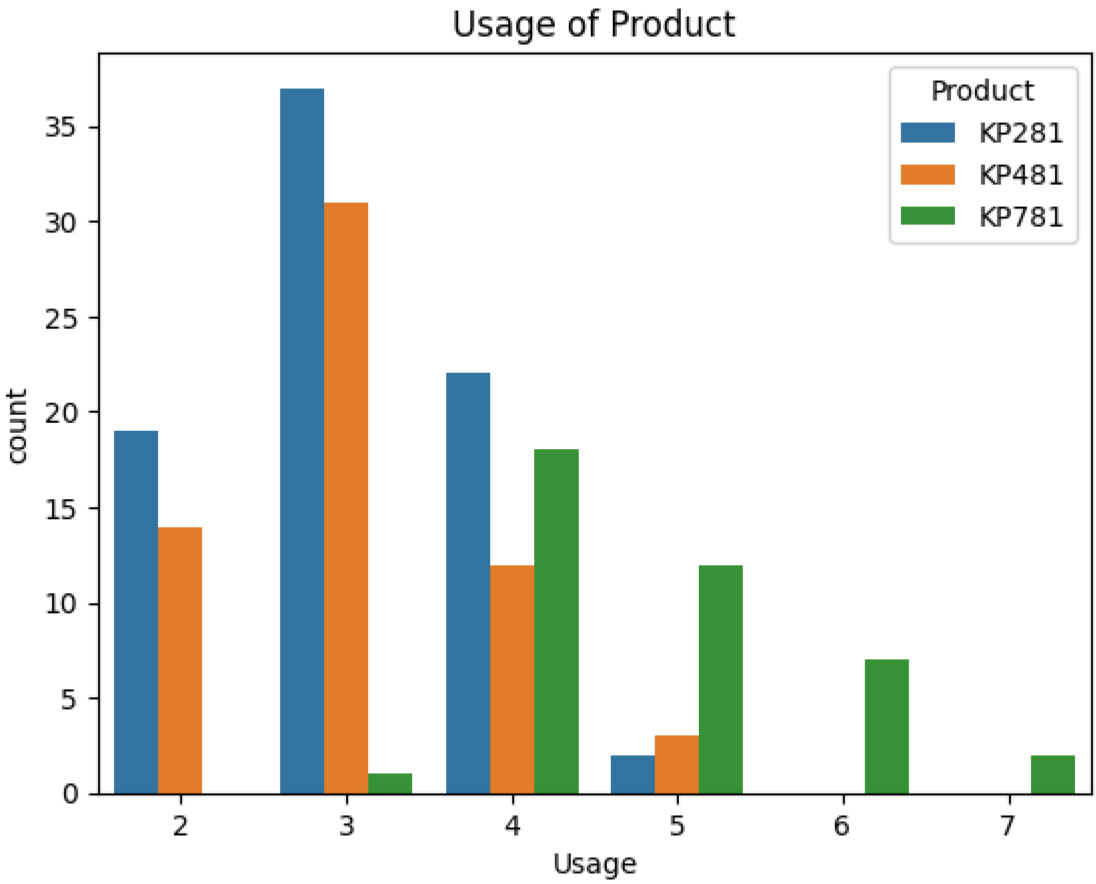
In [ ]:
```
# Analysing the rating for each individual

sns.kdeplot(data=data , x="Fitness", hue="Gender")

plt.title("Rating for each Individual")
plt.show()
```




**Insight :**

- From the above graph we can analyze that majority of Male and Female rated themselves as 3.

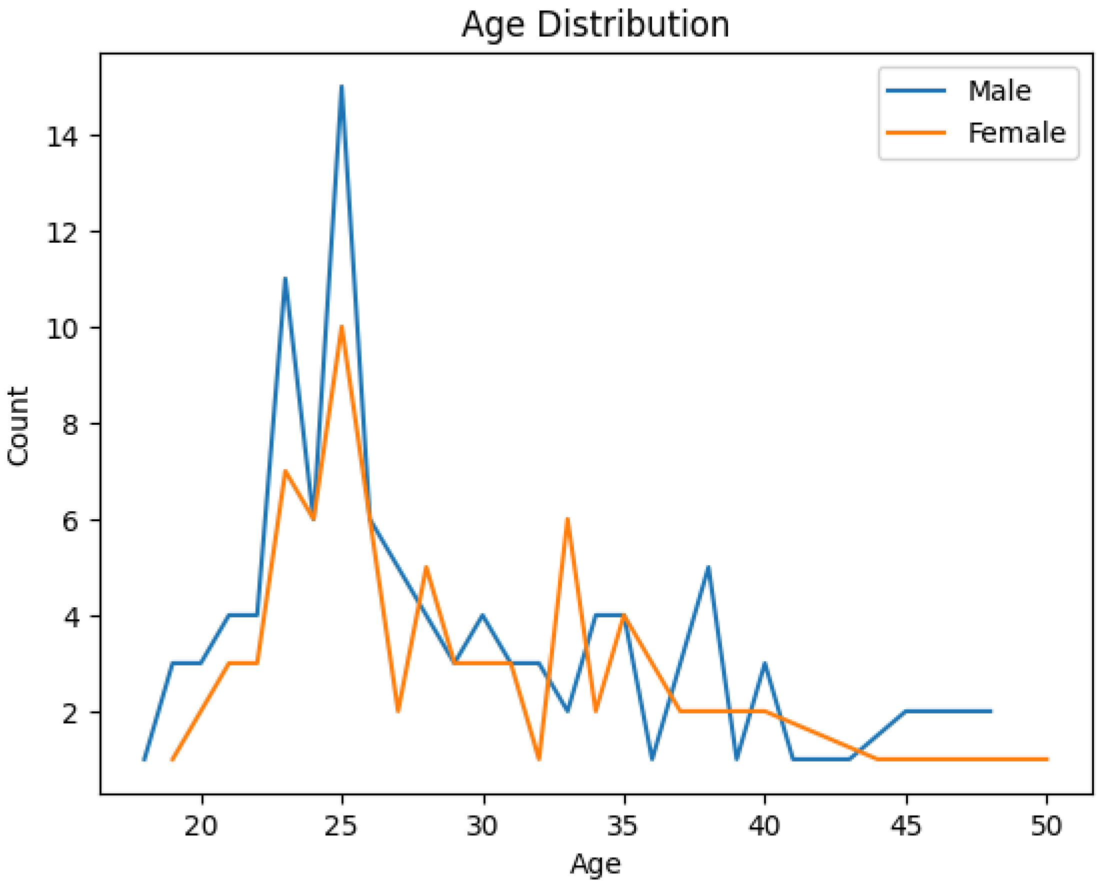
In [ ]:
```
# Analysing the average usage of the product

sns.countplot(x = "Usage" ,hue="Product",data=data)
plt.title("Usage of Product")

plt.show()
```




**Insight :**

- Compared to all the 3 products the most used product is KP281 where an average of 3 is the most used number of times.

In [ ]:
```
male = data[data["Gender"]=="Male"]
female = data[data["Gender"]=="Female"]

m_age = male["Age"] .value_counts() .sort_index()
f_age = female["Age"].value_counts().sort_index()

sns.lineplot(x = m_age.index , y = m_age , label = "Male")
sns.lineplot(x = f_age.index , y = f_age , label = "Female")

plt.xlabel("Age")
plt.ylabel("Count")

plt.title("Age Distribution")
plt.show()
```

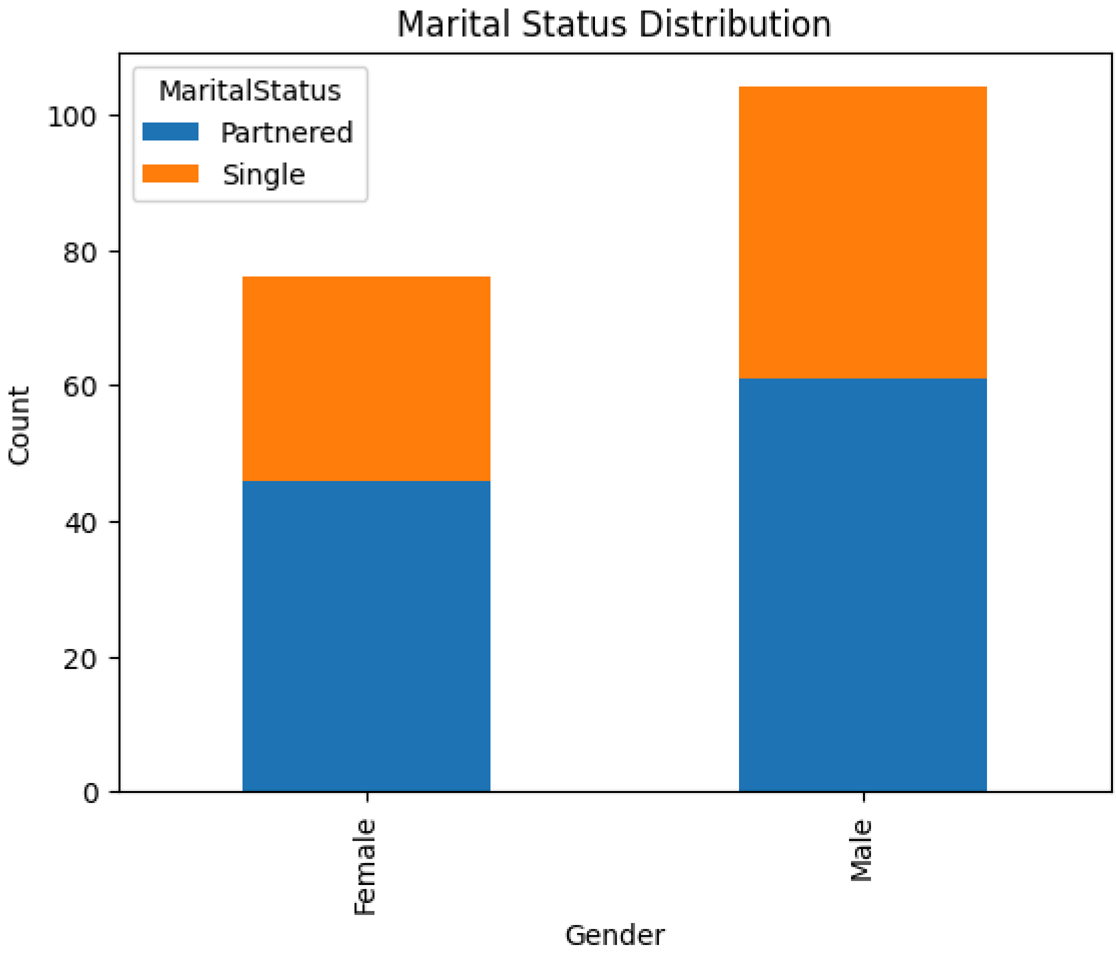



**Insight :**

- It is very eveident that male and female are of variable age where most of the male and female are of age 25.

In [ ]:
```
plt.figure(figsize=(3,8))

plot = pd.crosstab(index = data["Gender"] , columns = data["MaritalStatus"])

plot.plot(kind = "bar" , stacked = True)

plt.xlabel("Gender")
plt.ylabel("Count")

plt.title("Marital Status Distribution")
plt.show()
```

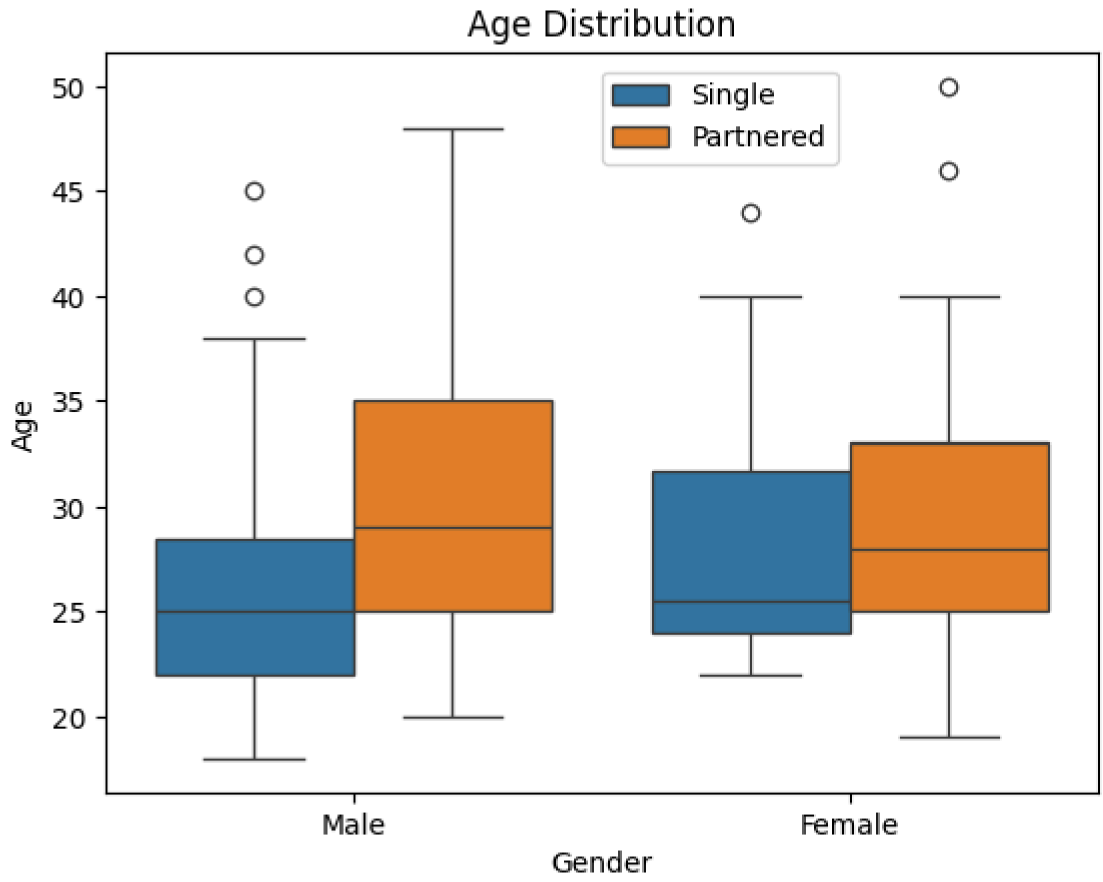
```
<Figure size 300x800 with 0 Axes>
```




**Insight :**

- From the above graph compared to female more male are parterned and single.

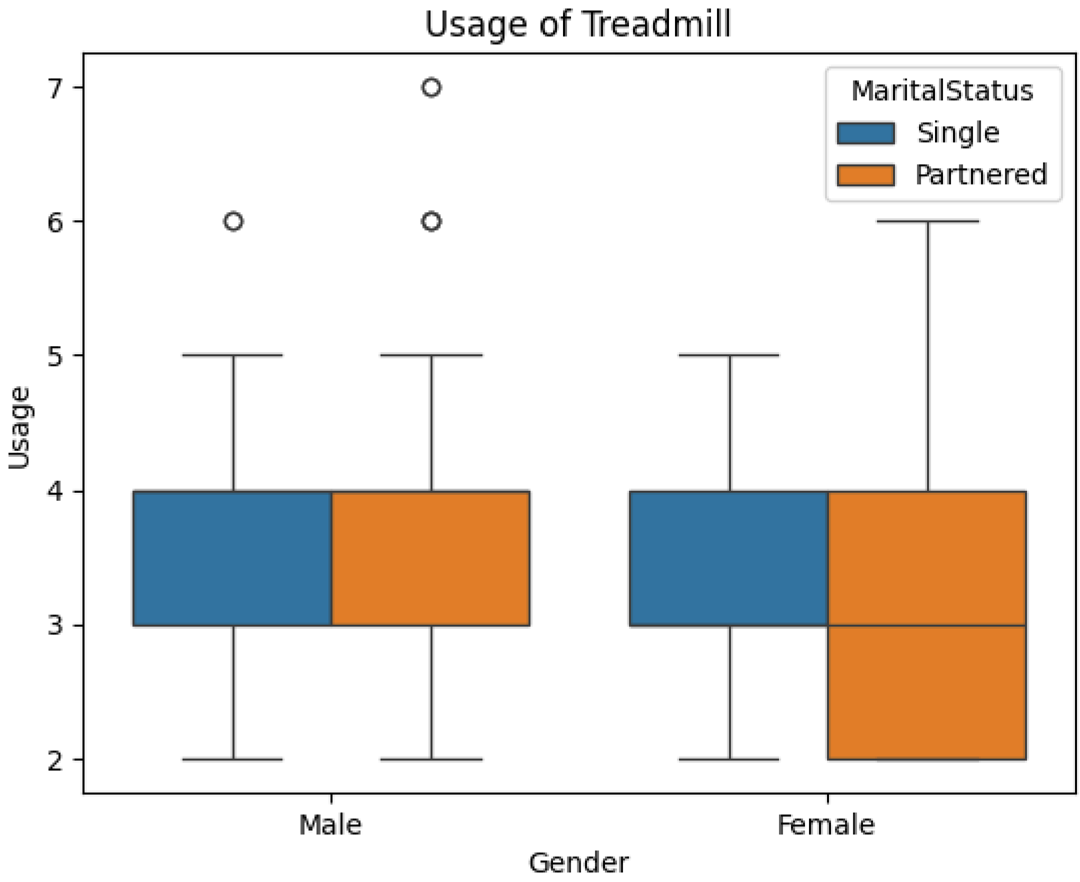
In [ ]:
```
sns.boxplot(y="Age",x="Gender" ,hue="MaritalStatus",  data = data)
plt.legend(loc=(0.5,0.85))

plt.title("Age Distribution")
plt.show()
```




**Insight :**

- In male the median age of Single is around 25 with 3 outliers whereas for parterned it is around 30.
- In female the median age for singles is around 25 with 1 outlier whereas parterned is around 27 with 2 outliers.

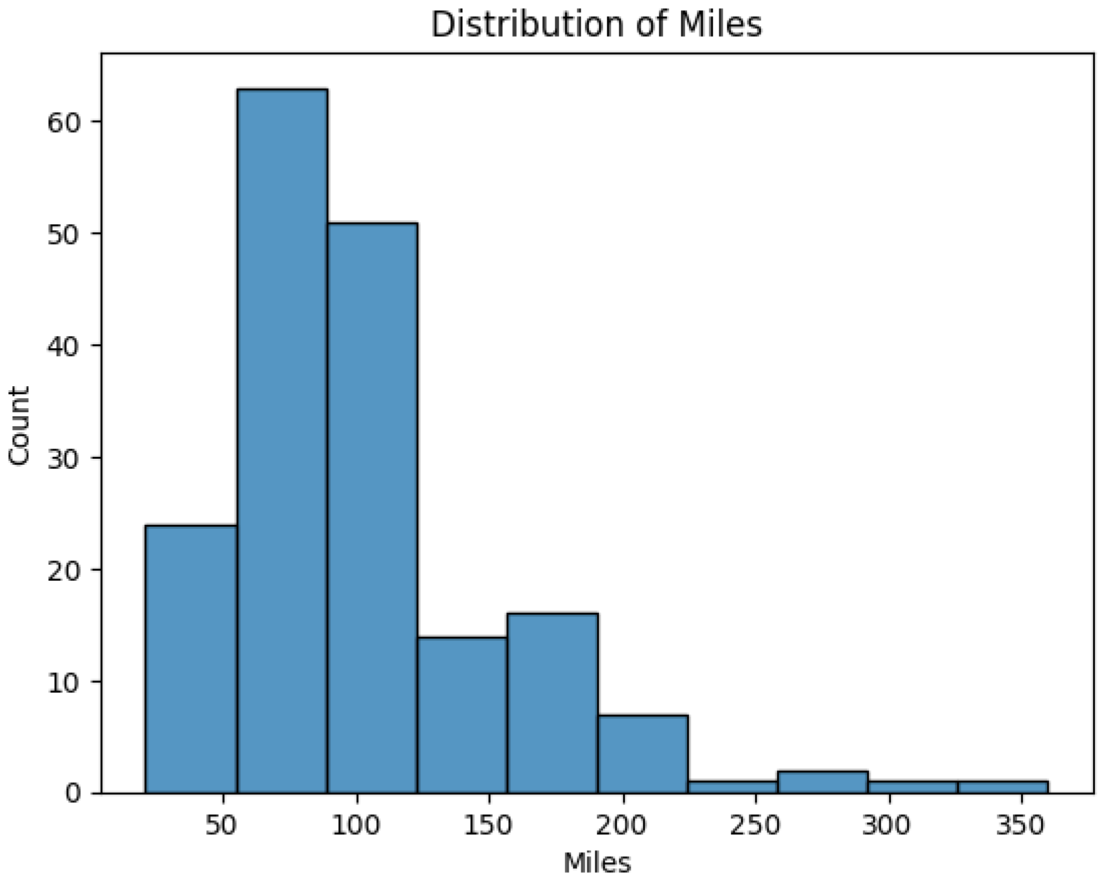
In [ ]:
```
sns.boxplot(y="Usage",x="Gender" ,hue="MaritalStatus",  data = data)

plt.title("Usage of Treadmill")
plt.show()
```




**Insight :**

- In male the median usage for both Single and partnered is around 4 with 1 and 2 outlier respectively.
- In female the median usage for both singles and partnered is around 3 .

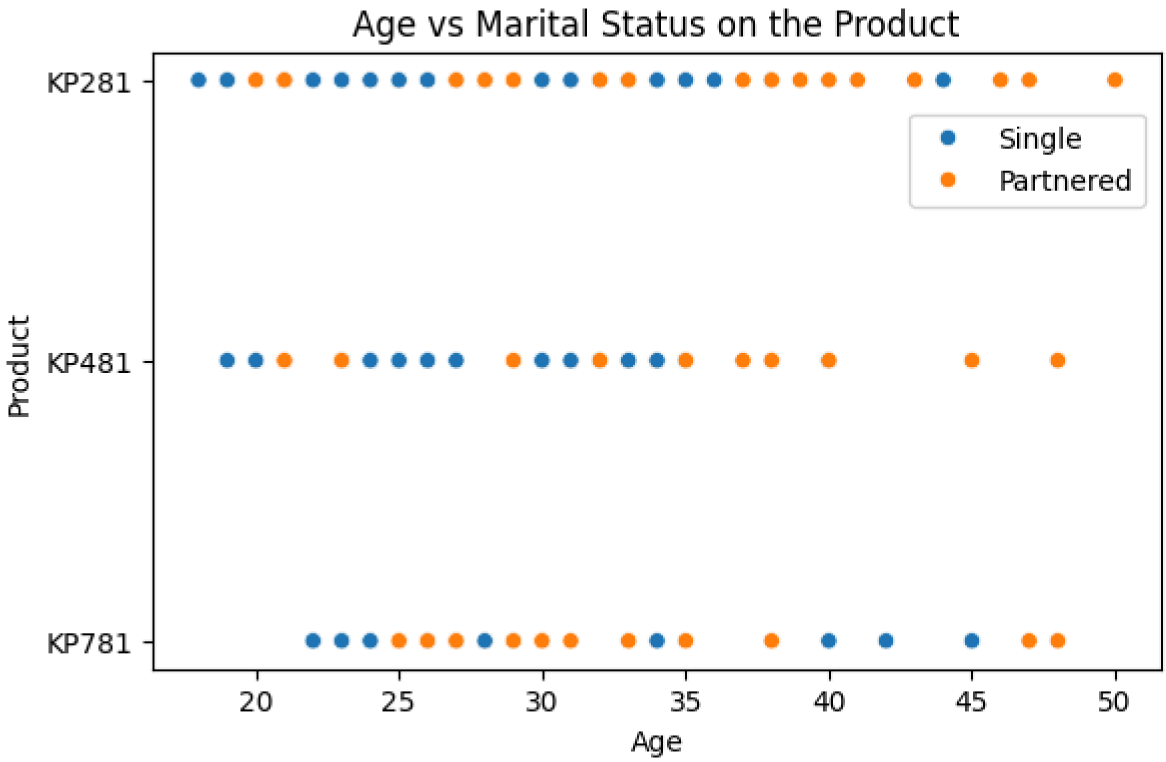
In [ ]:
```
sns.histplot(data["Miles"],bins=10)

plt.title("Distribution of Miles")
plt.show()
```



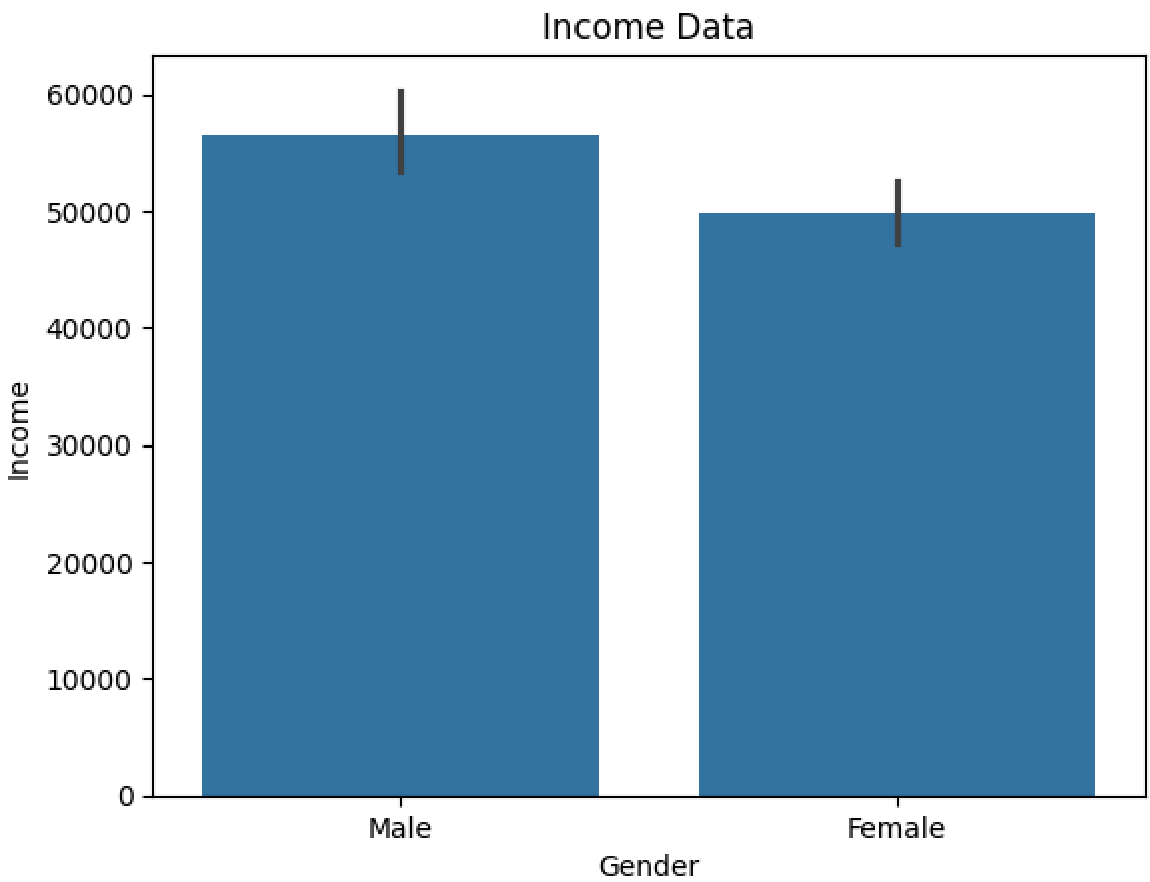

**Insight :**

- From the above graph its clear that most of the people walked between 50 and 100 miles.

In [ ]:
```
plt.figure(figsize = (6.5,4))

sns.scatterplot(data = data , y= "Product" , hue = "MaritalStatus", x ="Age")
plt.legend(loc=(0.75,0.75))

plt.title("Age vs Marital Status on the Product")
plt.show()
```

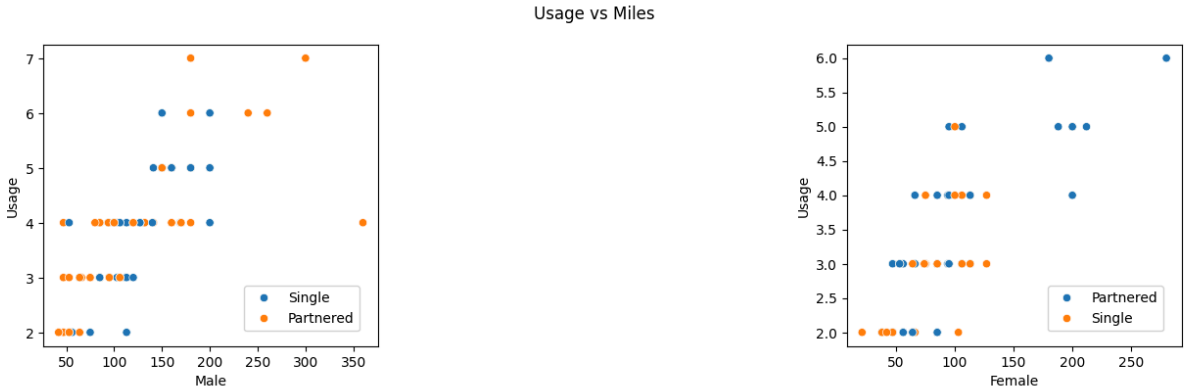



In [ ]:
```
sns.barplot(data=data , x = "Gender" , y = "Income" , estimator = np.mean)

plt.title("Income Data")
plt.show()
```




In [ ]:
```
male = data[data["Gender"]=="Male"]
female = data[data["Gender"]=="Female"]

plt.figure(figsize = (15,4)).suptitle("Usage vs Miles")
```

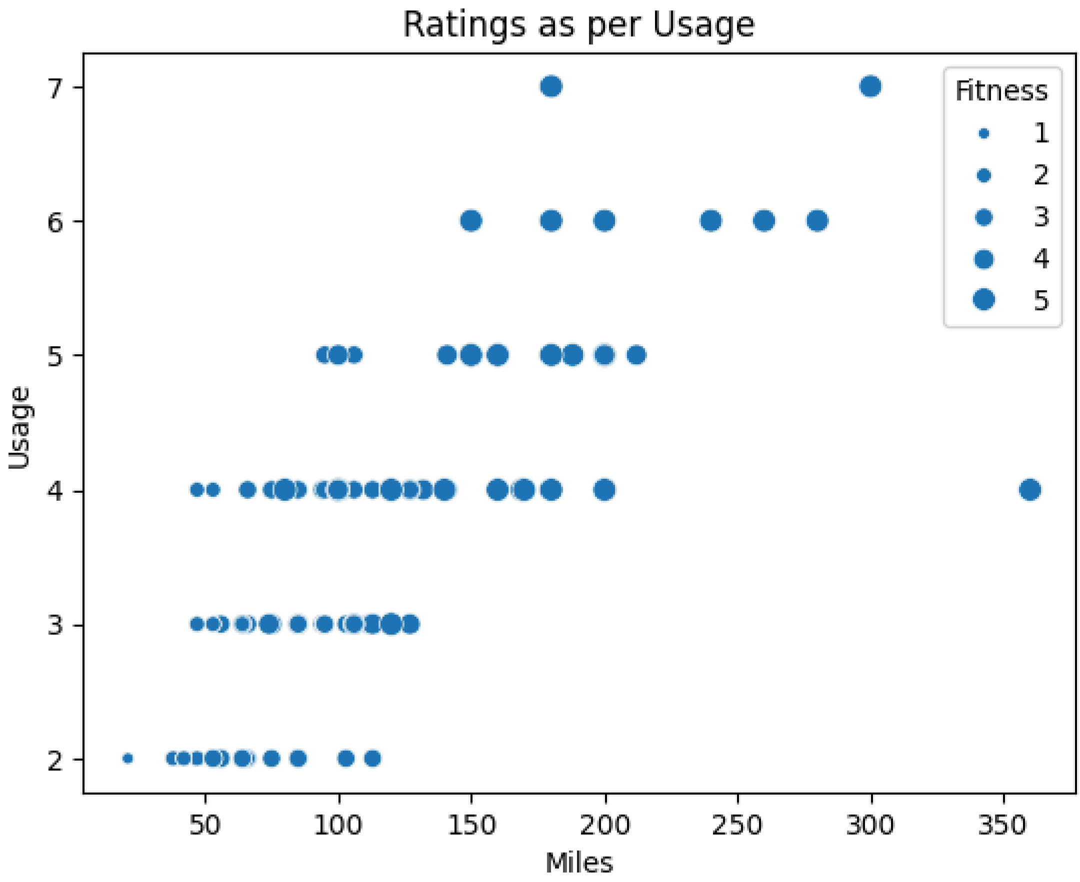
```
plt.subplot(1,3,1)
sns.scatterplot(data=male , x="Miles" , y ="Usage",hue="MaritalStatus")
plt.xlabel("Male")
plt.legend(loc=(0.6,0.05))

plt.subplot(1,3,3)
sns.scatterplot(data=female , x="Miles" , y ="Usage",hue="MaritalStatus")
plt.xlabel("Female")
plt.legend(loc=(0.6,0.05))


plt.show()
```




**Insight :**

- In male more miles is walked by parterned men whereas most of the men have walked in the range of 50 to 200 miles.
- In female most miles is walked by single women whereas most female walked in the range of 50 to 150 miles.

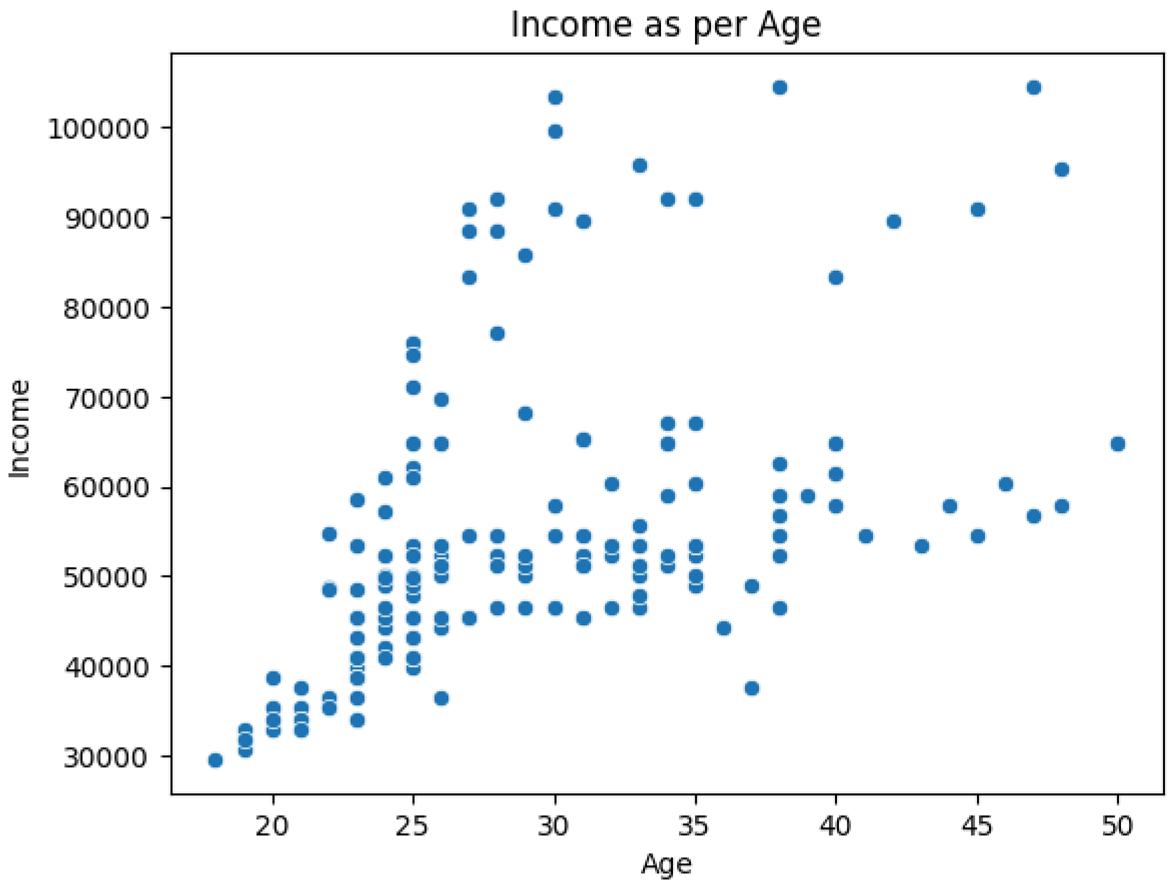
In [ ]:
```
sns.scatterplot(data = data , x="Miles"  , y="Usage" , size ="Fitness")

plt.title("Ratings as per Usage")
plt.show()
```




**Insight :**

- From the above graph its clear that more the usage is more the miles coved hence more the rating given.

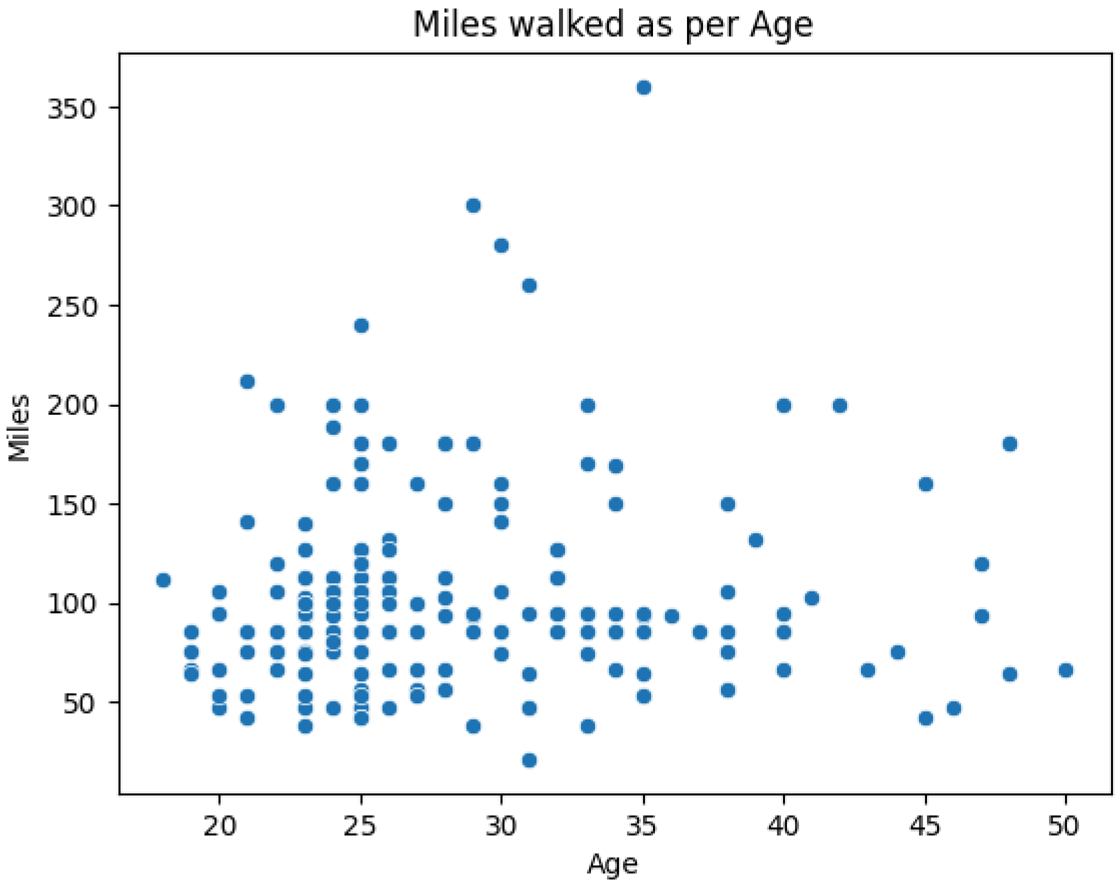
In [ ]:
```
sns.scatterplot(data = data , x="Age"  , y="Income")

plt.title("Income as per Age")
plt.show()
```




**Insight :**

- It is clear that Income is directly proportional to the Age, as the age increases the income also increases.

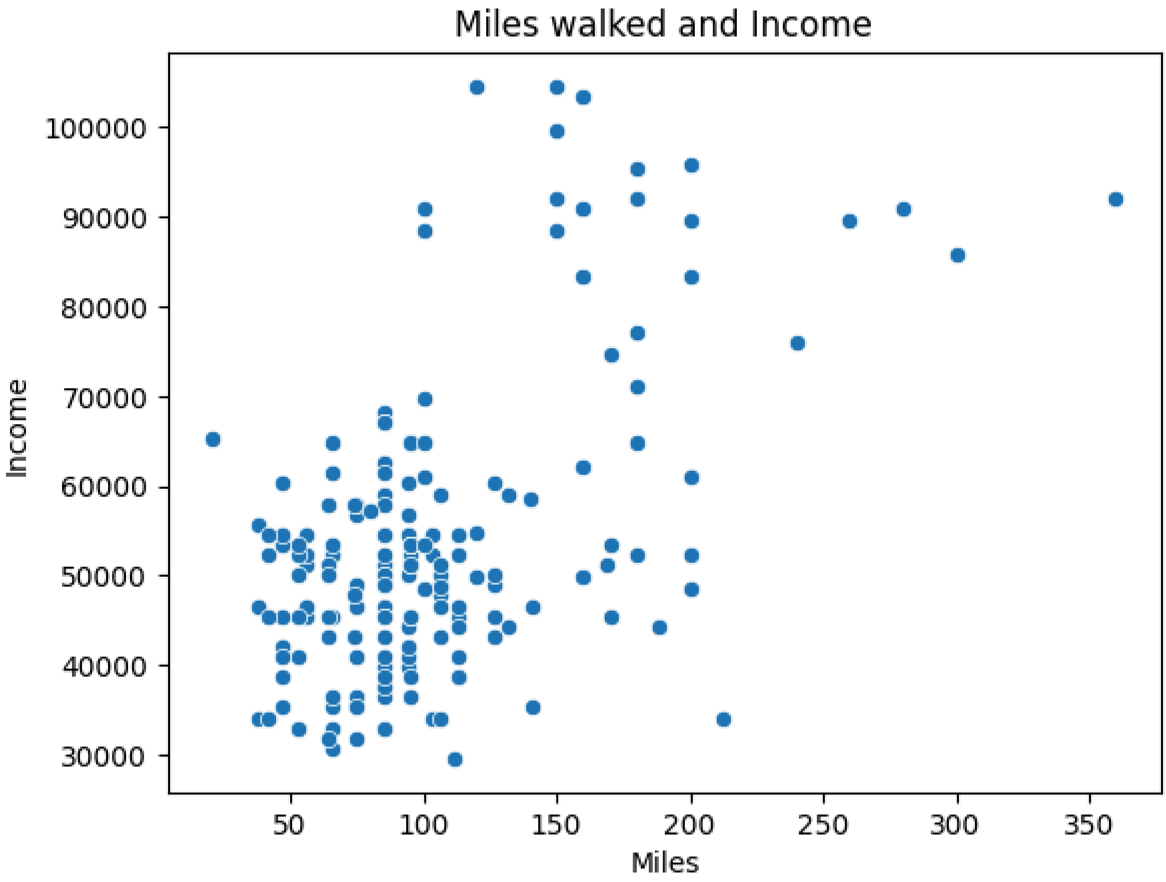
In [ ]:
```
sns.scatterplot(data = data , x="Age"  , y="Miles")

plt.title("Miles walked as per Age")
plt.show()
```




**Insight :**

- People in lower age has greater miles of walk where as the older people are having less miles of walk which needs to be improved for their better health.

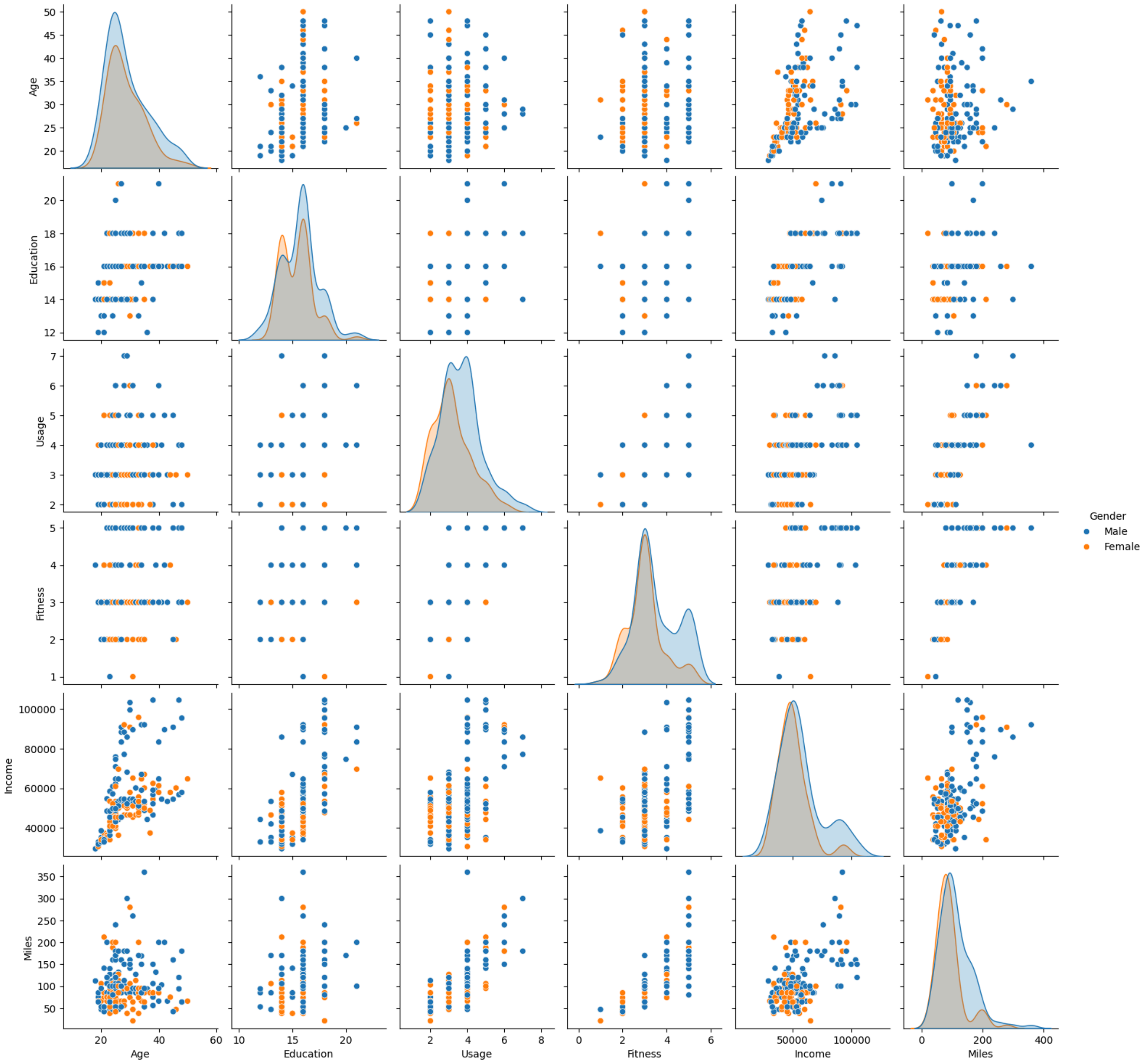
In [ ]:
```
sns.scatterplot(data = data , x="Miles"   , y="Income")

plt.title("Miles walked and Income")
plt.show()
```




**Insight :**

- People having greater salary have walked more miles compared to the people with lower income.

In [ ]:
```
sns.pairplot(data = data , hue = "Gender")
plt.show()
```




## Probability and Statistical Data

In [ ]:
```
#Probability that a male customer bought a treadmill

male = data.loc[data["Gender"]=="Male"]
prob_m = male.shape[0]/data.shape[0]
print(f"Probability of a male buying the product is {prob_m}")
prob_f = 1 - prob_m
print(f"Probability of a female buying the product is {prob_f}")
```

```
Probability of a male buying the product is 0.5777777777777777
Probability of a female buying the product is 0.4222222222222223
```

In [ ]:
```
detail = pd.crosstab(index = data["Gender"],
            columns = data["Product"],
            margins = True)
detail
```

Out[ ]:

| Product | KP281 | KP481 | KP781 | All |
|---|---|---|---|---|
| **Gender** | | | | |
| **Female** | 40 | 29 | 7 | 76 |
| **Male** | 40 | 31 | 33 | 104 |
| **All** | 80 | 60 | 40 | 180 |

### Male

- Probability of Male buying KP281 = P(Male | KP281) = 40/80 = 0.5
- Probability of Male buying KP481 = P(Male | KP481) = 31/60 = 0.516
- Probability of Male buying KP781 = P(Male | KP781) = 33/40 = 0.825

### Female

- Probability of Female buying KP281 = P(Female | KP281) = 40/80 = 0.5
- Probability of Female buying KP481 = P(Female | KP481) = 29/60 = 0.483
- Probability of Female buying KP781 = P(Female | KP781) = 7/40 = 0.175

In [ ]:
```
male = data.loc[data["Gender"]=="Male"]

detail_male = pd.crosstab(index = male["MaritalStatus"],
            columns = data["Product"],
            margins = True)
detail_male
```

Out[ ]:

| Product | KP281 | KP481 | KP781 | All |
|---|---|---|---|---|
| **MaritalStatus** | | | | |
| **Partnered** | 21 | 21 | 19 | 61 |
| **Single** | 19 | 10 | 14 | 43 |
| **All** | 40 | 31 | 33 | 104 |

In [ ]:
```
female = data.loc[data["Gender"]=="Female"]

detail_female = pd.crosstab(index = female["MaritalStatus"],
            columns = data["Product"],
            margins = True)
detail_female
```

Out[ ]:

| Product | KP281 | KP481 | KP781 | All |
|---|---|---|---|---|
| **MaritalStatus** | | | | |
| **Partnered** | 27 | 15 | 4 | 46 |
| **Single** | 13 | 14 | 3 | 30 |
| **All** | 40 | 29 | 7 | 76 |

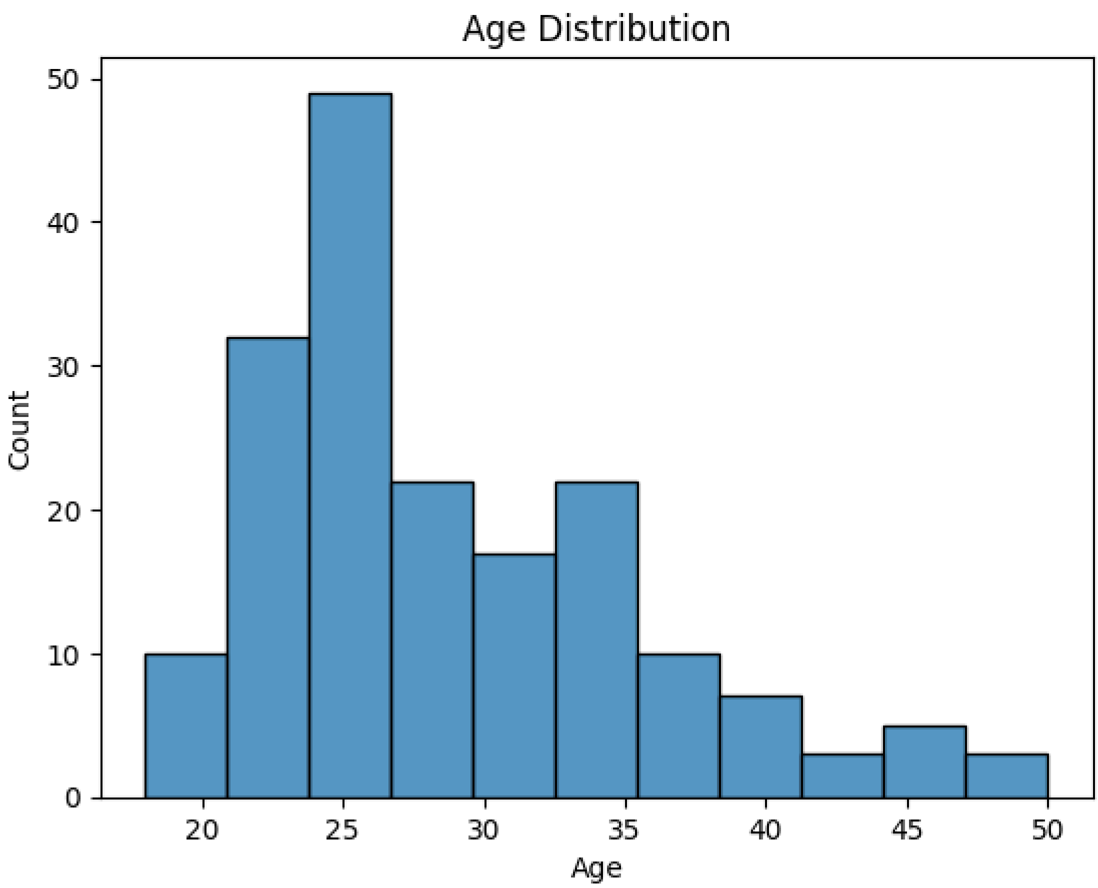
In [ ]:
```
#Analysing the Age Column

age = data["Age"]
age.describe()
```

Out[ ]:

| | Age |
|---|---|
| **count** | 180.000000 |
| **mean** | 28.788889 |
| **std** | 6.943498 |
| **min** | 18.000000 |
| **25%** | 24.000000 |
| **50%** | 26.000000 |
| **75%** | 33.000000 |
| **max** | 50.000000 |

**dtype:** float64

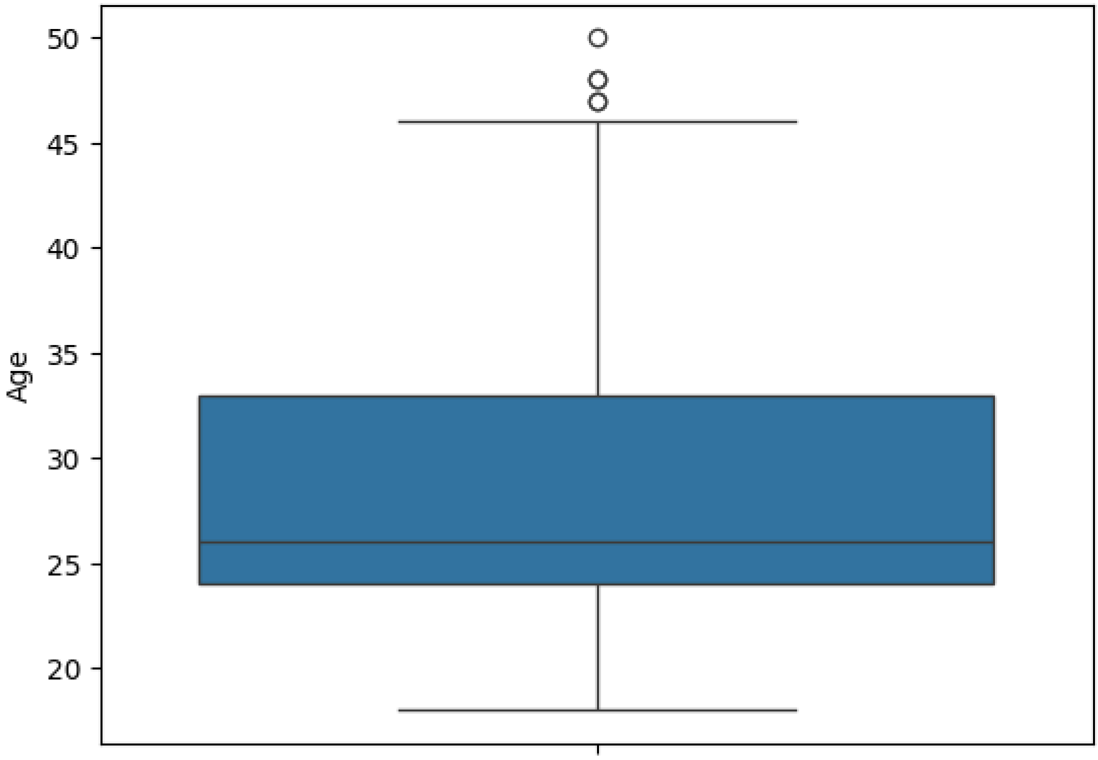
In [ ]:
```
sns.histplot(age)

plt.title("Age Distribution")
plt.show()
```




In [ ]:
```
percentile_25 = np.percentile(data["Age"],25)
print(f" 25% of the age is less than {percentile_25}")

percentile_50 = np.percentile(data["Age"],50)
print(f" 50% of the age is less than {percentile_50}")

percentile_75 = np.percentile(data["Age"],75)
print(f" 75% of the age is less than {percentile_75}")
```

```
25% of the age is less than 24.0
50% of the age is less than 26.0
75% of the age is less than 33.0
```

In [ ]:
```
sns.boxplot(data["Age"])

plt.show()
```




In [ ]:
```
#Range

range_ = (data["Age"].max() - data["Age"].min())
print(f"The range for Age is {range_}")
```

```
The range for Age is 32
```

In [ ]:
```
#Mean

mean_ = np.mean(data["Age"])
print(f"The mean for Age is {mean_}")
```

```
The mean for Age is 28.788888888888888
```

In [ ]:
```
#Variance

var_ = np.var(data["Age"])
print(f"The Variance of Age is {var_}")
```

```
The Variance of Age is 47.94432098765432
```

In [ ]:
```
#Standard Deviation

std_dev_ = np.std(data["Age"])
print(f"The standard deviation of Age is {std_dev_}")
```

```
The standard deviation of Age is 6.924183777720976
```

In [ ]:
```
# Comparison Between Male and Female Age

male = data[data["Gender"]=="Male"]
female = data[data["Gender"]=="Female"]
```

```
male_miles = male["Miles"]
female_miles = female["Miles"]
```

### Male

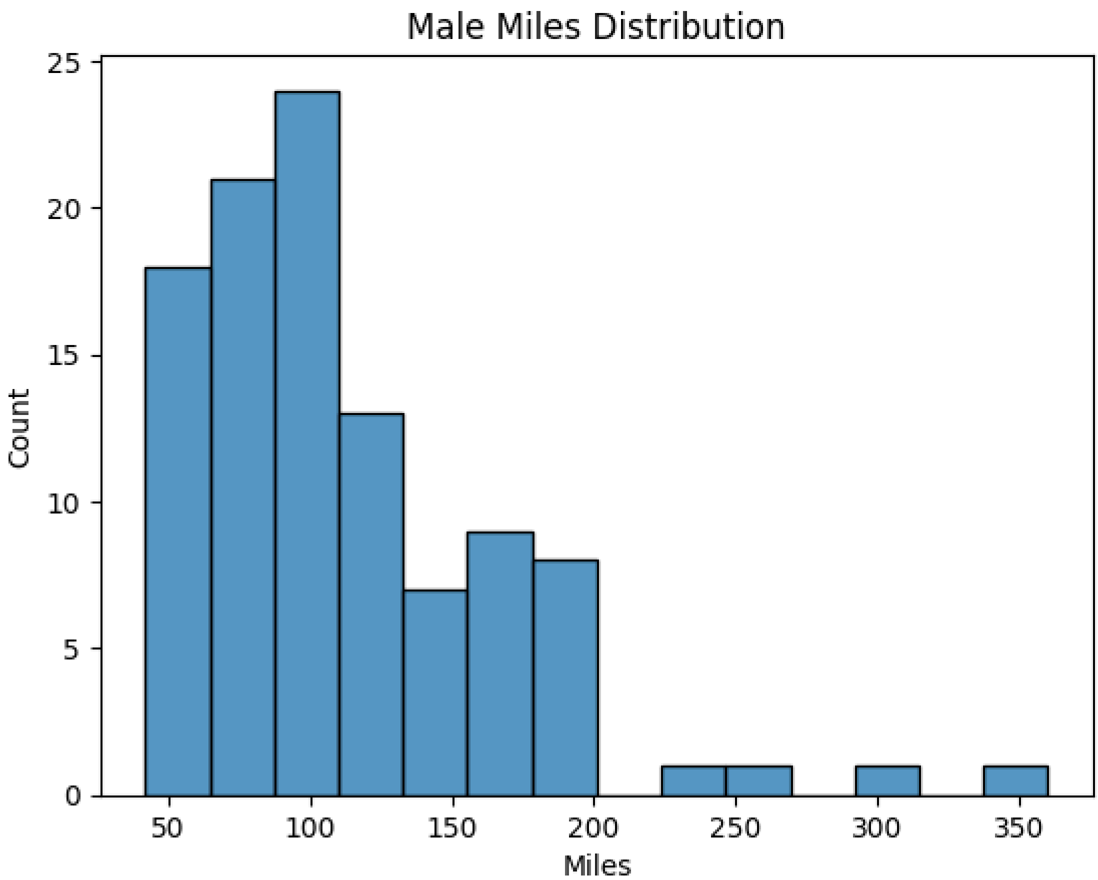
In [ ]:
```
male_miles.describe()
```

Out[ ]:

| | Miles |
|---|---|
| **count** | 104.000000 |
| **mean** | 112.826923 |
| **std** | 54.702451 |
| **min** | 42.000000 |
| **25%** | 85.000000 |
| **50%** | 100.000000 |
| **75%** | 141.000000 |
| **max** | 360.000000 |

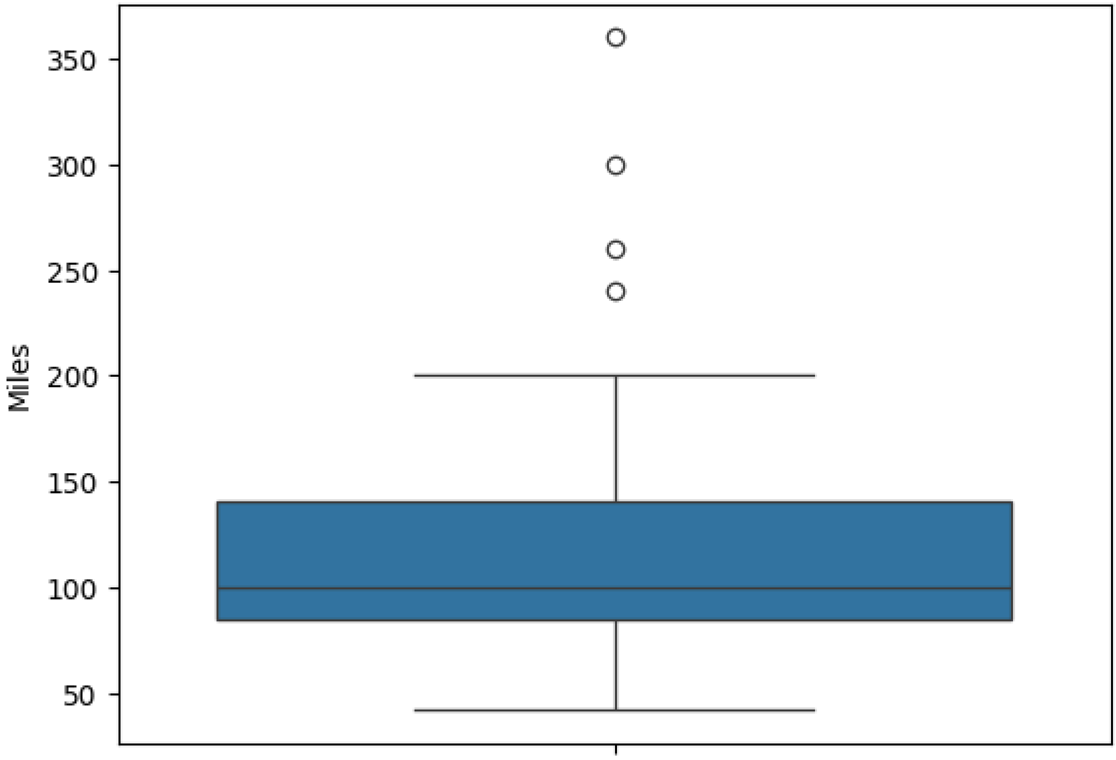
**dtype:** float64

In [ ]:
```
sns.histplot(male_miles)

plt.title("Male Miles Distribution")
plt.show()
```




In [ ]:
```
sns.boxplot(male["Miles"])
```

Out[ ]: <Axes: ylabel='Miles'>




In [ ]:
```
#Range

range_ = (male["Miles"].max() - male["Miles"].min())
print(f"The range for Male Miles is {range_}")
```

The range for Male Miles is 318

In [ ]:
```
#Mean

mean_ = np.mean( male["Miles"])
print(f"The mean for Male Miles is {mean_}")
```

The mean for Male Miles is 112.82692307692308

In [ ]:
```
#Variance

var_ = np.var(male["Miles"])
print(f"The Variance of Male Miles is {var_}")
```

The Variance of Male Miles is 2963.585428994083

In [ ]:
```
#Standard Deviation

std_dev_ = np.std(male["Miles"])
print(f"The standard deviation of Male Miles is {std_dev_}")
```

The standard deviation of Male Miles is 54.438822810509805

**Female**

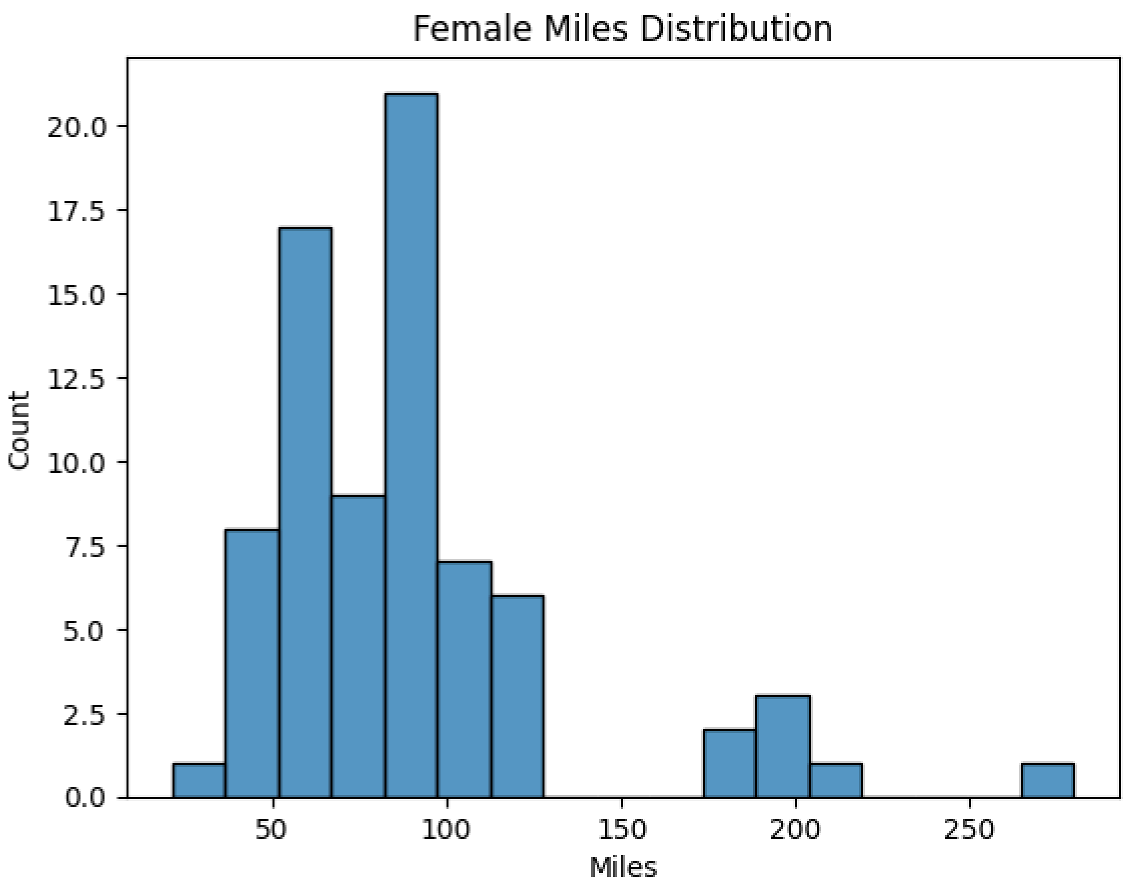
In [ ]:
```
female_miles.describe()
```

Out[ ]:

| | Miles |
|---|---|
| **count** | 76.000000 |
| **mean** | 90.013158 |
| **std** | 44.782882 |
| **min** | 21.000000 |
| **25%** | 66.000000 |
| **50%** | 85.000000 |
| **75%** | 100.000000 |
| **max** | 280.000000 |

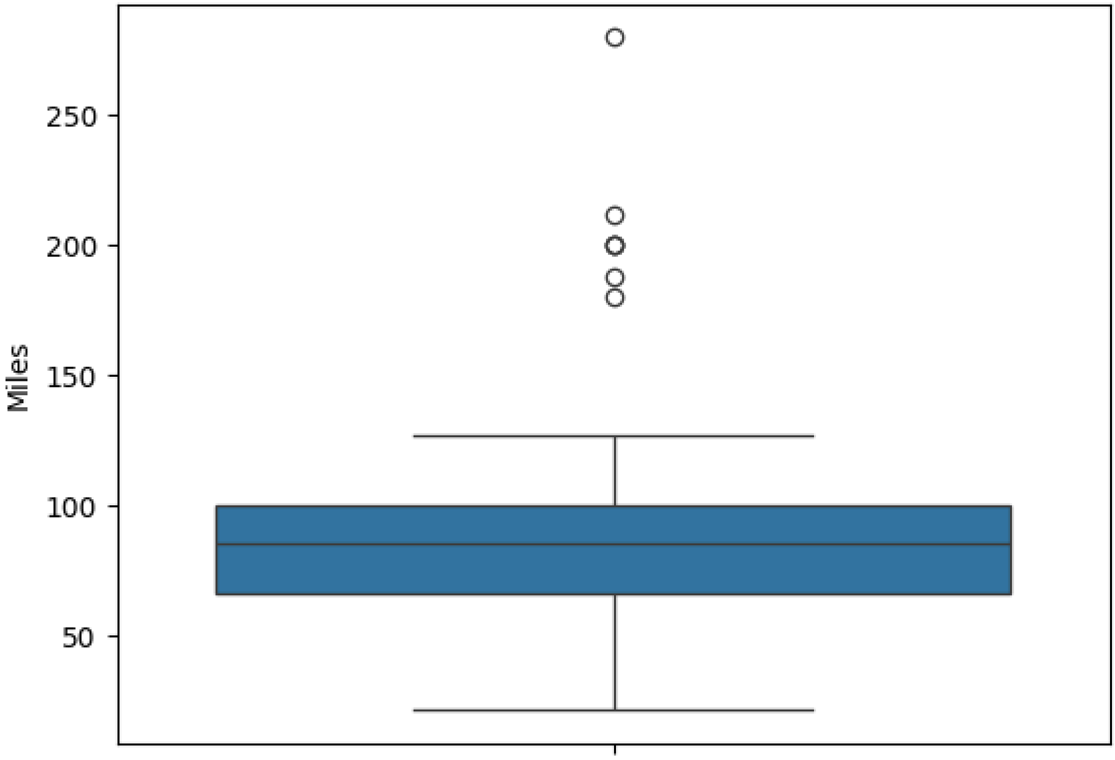
**dtype:** float64

In [ ]:
```
sns.histplot(female_miles)

plt.title("Female Miles Distribution")
plt.show()
```




In [ ]:
```
sns.boxplot(female["Miles"])
```

Out[ ]:
```
<Axes: ylabel='Miles'>
```




In [ ]:
```
#Range

range_ = (female["Miles"].max() - female["Miles"].min())
print(f"The range for Female Miles is {range_}")
```

```
The range for Female Miles is 259
```

In [ ]:
```
#Mean

mean_ = np.mean( female["Miles"])
print(f"The mean for Female Miles is {mean_}")
```

```
The mean for Female Miles is 90.01315789473684
```

In [ ]:
```
#Variance

var_ = np.var(female["Miles"])
print(f"The Variance of Female Miles is {var_}")
```

```
The Variance of Female Miles is 1979.1182479224376
```

In [ ]:
```
#Standard Deviation

std_dev_ = np.std(female["Miles"])
print(f"The standard deviation of Female Miles is {std_dev_}")
```

```
The standard deviation of Female Miles is 44.487281867095874
```

#### Conclusion

Std Dev of Male - 54.43

Std Dev of Female - 44.48

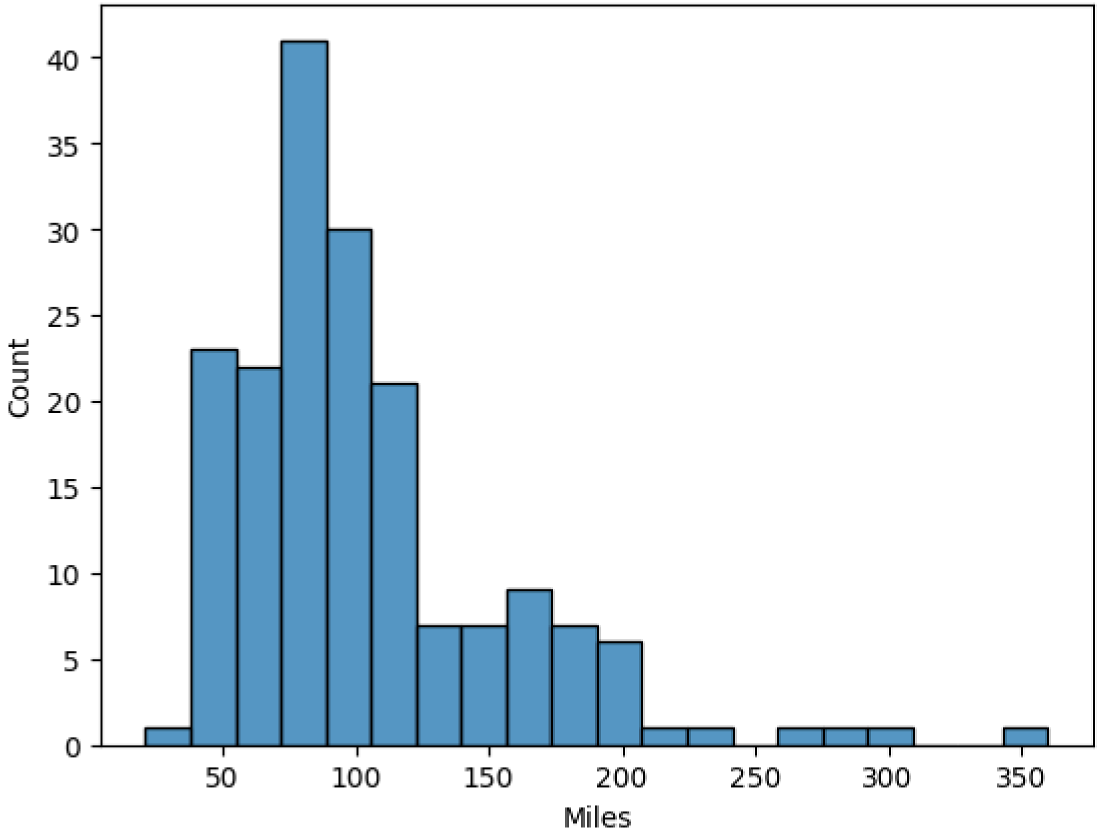
Females tend to be more stable and consistent in their performace compared to Males whereas Males tend to show more variance in walking.

#### Central Limit Theorem

In [ ]:
```
#Analysing the Entire population

miles = data["Miles"]
```

In [ ]:
```
sns.histplot(miles)
```

Out[ ]: <Axes: xlabel='Miles', ylabel='Count'>




In [ ]:
```
# Mean of the population

mu = data["Miles"].mean()
print(f"Mean of the entire population is {mu}")
```

```
Mean of the entire population is 103.19444444444444
```

In [ ]:
```
# Std Dev of the population

sigma = data["Miles"].std()
print(f"Standard Deviation of the entire population is {sigma}")
```

```
Standard Deviation of the entire population is 51.86360466180931
```

In [ ]:
```
# Taking a sample of size 15

sample_ = miles.sample(15)
sample_
```

Out[ ]:

| | Miles |
|---|---|
| **16** | 103 |
| **30** | 85 |
| **133** | 85 |
| **60** | 85 |
| **53** | 141 |
| **32** | 47 |
| **88** | 85 |
| **67** | 85 |
| **147** | 80 |
| **125** | 95 |
| **145** | 100 |
| **176** | 200 |
| **73** | 66 |
| **23** | 188 |
| **40** | 85 |

**dtype:** int64

In [ ]:
```
# Mean of the sample data

x =np.mean(sample_)
print(f"Mean of the sample data is {x}")
```

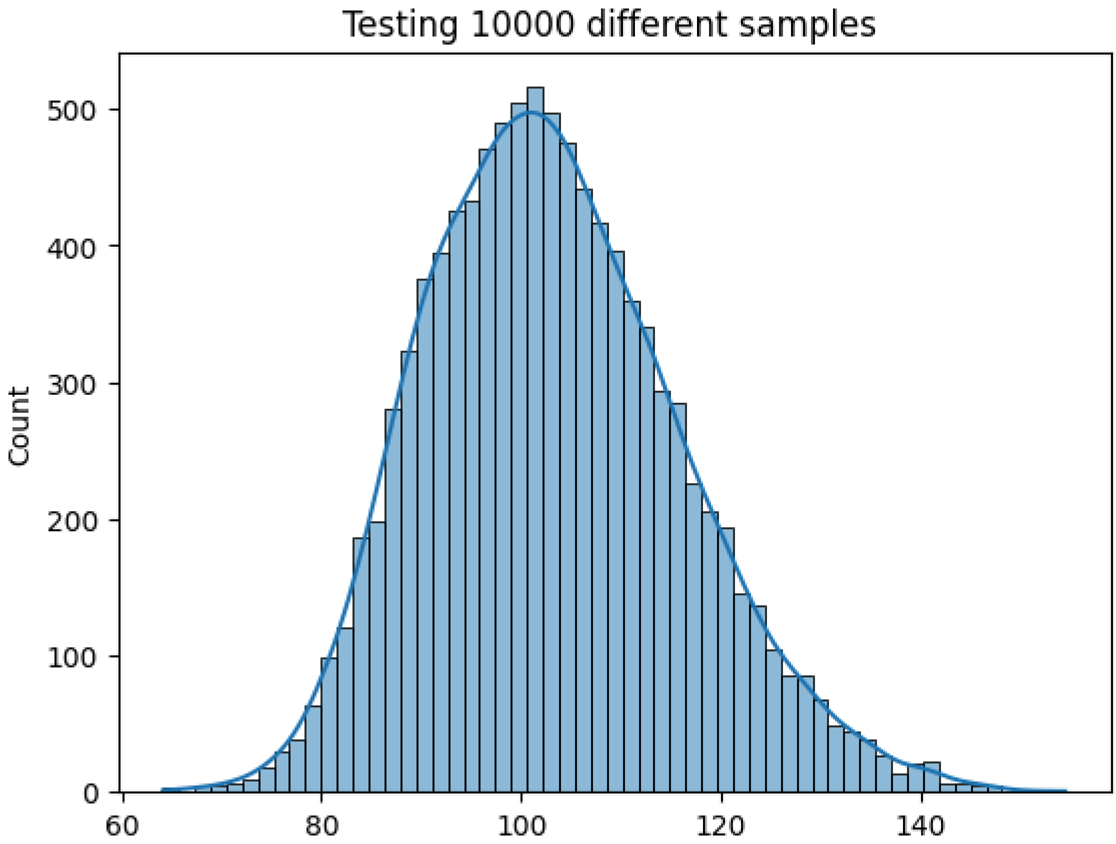
```
Mean of the sample data is 102.0
```

### Insight

It is clearly seen mean changes everytime the sample is executed as the sample data keeps changing.

In [ ]:
```
# Repeating the sampling process for 10000 times

sample_test = [np.mean(miles.sample(15)) for i in range(10000)]
```

In [ ]:
```
sns.histplot(sample_test , kde = True)

plt.title("Testing 10000 different samples ")
plt.show()
```




**Insight :**

From the graph its clearly seen that most of the data lies between 98 to 105.

Here the data ranges between 65 to 157.

In [ ]:
```
#Computing the Sample Mean

print(f"Mean of the sample tested population is {np.mean(sample_test)}")
```

```
Mean of the sample tested population is 103.12035333333333
```

In [ ]:
```
#Computing the Sample Standard deviation

print(f"Standard Deviation of the sample tested population is {np.std(sample_test)}
```

```
Standard Deviation of the sample tested population is 12.712480690147686
```

**Insight :**

The above operation proves that the population mean is almost same as that of the sample mean.

The stand deviation decreases as the size of the sample increases.

**Recommendation :**

- As the product KP281 is bought the most , more offers can be provided and the other products can be promoted more as the sales for others too can be increased.
- More competitions can be organised so that people get motivated and the average mile covered and the usage gets increased.

- The female users are comparitively less compared to male hence schemes can be introduced to cover the female users too.
- The parterned customers are comparitvely low to the single customers , parterned challenges and competitions can be encouraged.
- Majority customers are youths between 25 to 35 , whereas the ages customers are low , hence more awareness regarding the health can be spread among the people of age above 40.
- The average use of treadmill a week is 3 , hence more promotion can be done to increase the usage per week which in turn increases the mile coverd and the rating provided, at the end quality of health too gets improved.

In [ ]: